## CLAIMS

What is claimed is:

1. A method, comprising:

   in an object oriented environment:

   a) invoking a second method from a first method, said invoking comprising providing an identification of said first method and a class that said first method is a part of;

   b) identifying a plug-in module for said first method based upon said identification, said plug-in module containing a handler method;

   c) executing said handler method to perform a output function for said first method; and,

   d) executing said first method from a point beyond where said second method was invoked.

2. The method of claim 1 wherein said executing of said handler method causes an entry time for said first method to be recorded .

3. The method of claim 1 wherein said executing of said handler method causes an exit time for said first method to be recorded.

4. The method of claim 1 wherein said executing of said handler method causes a counter maintained for said first method to be incremented.

5. The method of claim 1 wherein said executing of said handler method causes an input parameter value of said first method to be recorded.

6. The method of claim 1 wherein said executing of said handler method causes a returned value of said first method to be recorded.

7. The method of claim 1 wherein said first method is a constructor.

8. The method of claim 1 further comprising creating, prior to said invoking, an object having an input parameter value of said first method.

9. The method of claim 1 wherein said invoking further comprises providing an input parameter value of said first method.

10. The method of claim 1 wherein said invoking further comprises identifying where said first method's instructions can be found in memory.

11. The method of claim 1 further comprising, after said executing a portion of said first method:

e) invoking a third method from said first method because said first method is about to reach an exit point, said second method having been invoked from said first method because an entry point of said first method had just been reached;

f) re-identifying said plug-in module for said first method as a consequence of said invoking a third method;

g) re-executing said handler method to perform a output function for said first method; and,

h) executing a remaining portion of said first method through said exit point.

12. The method of claim 1 further comprising, after said executing a portion of said first method:

e) flowing from said first method to a third method

f) invoking said second method from said third method, said invoking comprising providing an identification of said third method and a second class that said third method is a part of;

g) identifying said plug-in module for said third method based upon said third method and second class identification;

h) executing said handler method to perform said output function for said third method; and,

i) executing a portion of said third method from a point beyond where said second method was invoked.

13. The method of claim 12 further comprising also identifying a second plug-in module for said third method based upon said third method and second class identification, said second plug-in module containing a second handler method.

14. The method of class 13 further comprising also executing said second handler method to perform a different output function than said output function for said third method

15. The method of claim 14 wherein a first object is called to execute said first method and a second method is called to execute said third object.

16. The method of claim 15 wherein said object oriented environment is a Java object oriented environment.

17. The method of claim 1 wherein said invoking further comprises providing said first method signature, said first method's signature comprising:

said identification of said first method;

said identification of said class that said first method is a part of; and,

said first method's arguments.

18. One or more machine readable media containing instructions which when executed by one or more computing systems cause a method to be performed, said method, comprising:

in an object oriented environment:

a) invoking a second method from a first method, said invoking comprising providing an identification of said first method and a class that said first method is a part of;

b) identifying a plug-in module for said first method based upon said identification, said plug-in module containing a handler method;

c) executing said handler method to perform a output function for said first method; and,

d) executing said first method from a point beyond where said second method was invoked.

19. The one or more machine readable media of claim 18 wherein said executing of said handler method causes an entry time for said first method to be recorded .

20. The one or more machine readable media of claim 18 wherein said executing of said handler method causes an exit time for said first method to be recorded.

21. The one or more machine readable media of claim 18 wherein said executing of said handler method causes a counter maintained for said first method to be incremented.

22. The one or more machine readable media of claim 18 wherein said executing of said handler method causes an input parameter value of said first method to be recorded.

23. The one or more machine readable media of claim 18 wherein said executing of said handler method causes a returned value of said first method to be recorded.

24. The one or more machine readable media of claim 18 wherein said first method is a constructor.

25. The one or more machine readable media of claim 18 further comprising creating, prior to said invoking, an object having an input parameter value of said first method.

26. The one or more machine readable media of claim 18 wherein said invoking further comprises providing an input parameter value of said first method.

27. The one or more machine readable media of claim 18 wherein said invoking further comprises identifying where said first method's instructions can be found in memory.

28. The one or more machine readable media of claim 18 further comprising, after said executing a portion of said first method:

e) invoking a third method from said first method because said first method is about to reach an exit point, said second method having been invoked from said first method because an entry point of said first method had just been reached;

f) re-identifying said plug-in module for said first method as a consequence of said invoking a third method;

g) re-executing said handler method to perform a output function for said first method; and,

h) executing a remaining portion of said first method through said exit point.

29. The one or more machine readable media of claim 18 further comprising, after said executing a portion of said first method:

e) flowing from said first method to a third method

f) invoking said second method from said third method, said invoking comprising providing an identification of said third method and a second class that said third method is a part of;

g) identifying said plug-in module for said third method based upon said third method and second class identification;

h) executing said handler method to perform said output function for said third method; and,

i) executing a portion of said third method from a point beyond where said second method was invoked.

30. The one or more machine readable media of claim 29 further comprising also identifying a second plug-in module for said third method based upon said third method and second class identification, said second plug-in module containing a second handler method.

31. The one or more machine readable media of class 30 further comprising also executing said second handler method to perform a different output function than said output function for said third method

32. The one or more machine readable media of claim 31 wherein a first object is called to execute said first method and a second method is called to execute said third object.

33. The one or more machine readable media of claim 32 wherein said object oriented environment is a Java object oriented environment.

34. The one or more machine readable media of claim 18 wherein said invoking further comprises providing said first method signature, said first method's signature comprising:

said identification of said first method;

said identification of said class that said first method is a part of; and, said first method's arguments.